

Resumen Teórico - Organización del Computador I

Dana Tilve

Índice

1. Arquitectura de Von Neumann	2
1.1. Stored-program vs Program-controlled	3
1.1.1. ISA (Instruction Set Architecture)	3
2. Input/Output	4
2.1. Hardware Interrupts	5
2.2. Software Interrupts	5
3. Buses	7
3.1. Diseño de Bus	7
3.1.1. Tipo de líneas	7
3.1.2. Ancho del Bus	8
3.1.3. Temporización	8
3.2. Arbitraje	8
3.3. Conexiones	9
4. Memorias	10
4.1. Tipos y tecnología de memorias	10
5. Memorias caché	12
5.1. Performance	12
5.1.1. Impacto de un caché miss	12
5.2. Controlador de caché	12
5.3. Coherencia	13
5.4. Estructura de memoria caché	13
5.4.1. Organización	13
5.4.2. Políticas de reemplazo	13
5.5. Implementaciones prácticas de memoria caché	13

1. Arquitectura de Von Neumann

La arquitectura de una máquina es la descripción de las capacidades y el modelo de programación de un ordenador (no la implementación particular). El diseño de la arquitectura puede implicar el diseño del conjunto de instrucciones, diseño de microarquitectura, diseño de la lógica, y su implementación.

La Arquitectura de Von Neumann, también conocida como modelo de Von Neumann o arquitectura Princeton, es aquella arquitectura de procesador basada en la descrita en 1945 por el matemático y físico John von Neumann que consiste en:

1. Una CPU (central processing unit) que contiene:

ALU (Arithmetic Logic Unit) es un circuito digital electrónico que realiza las operaciones aritméticas y lógicas bit a bit en números binarios enteros.

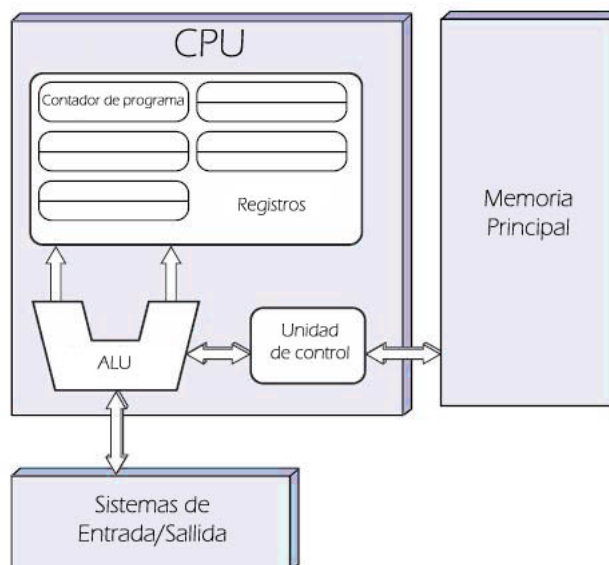
CU (Control Unit) es un componente que dirige las operaciones del procesador. Le dice a la memoria, ALU y los dispositivos de entrada y salida cómo responder a las instrucciones de un programa. Contiene a su vez:

IR almacena la instrucción que se está ejecutando actualmente o está siendo decodificada.

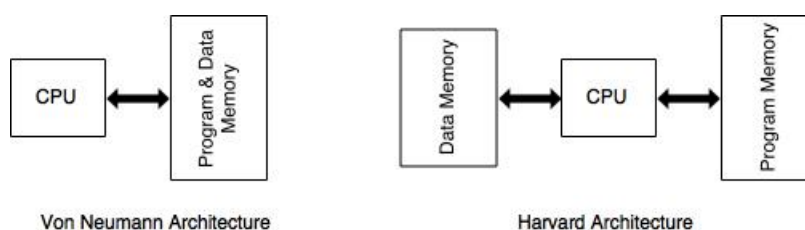
Registros son unidades de almacenamiento pequeñas que son típicamente dirigidas por mecanismos distintos de la memoria principal y a los que se puede acceder más rápido. Dentro de los SPR (special purpose registers) se encuentra:

PC indica en qué parte del programa está de la secuencia de un programa. Aumenta luego de hacer fetch de una instrucción.

2. Memoria (para almacenar tanto datos como instrucciones)
3. Almacenamiento masivo externo
4. Mecanismos de entrada y salida.



El diseño de la arquitectura Von Neumann es más simple que la arquitectura de Harvard, que tiene un conjunto dedicado de direcciones y buses de datos para leer datos desde memoria y escribir datos en la misma, y otro conjunto de direcciones y buses de datos para buscar instrucciones.



La limitación de rendimiento de esta arquitectura es que no pueden ocurrir una extracción de instrucción y una operación de datos al mismo tiempo, ya que comparten un bus en común. Esto se conoce como el cuello de botella Von Neumann. Además, por diseño o accidente, es posible modificar el código del programa lo cual puede resultar dañino para sí mismo, otros programas o el sistema operativo, lo que posiblemente derive en crash.

1.1. Stored-program vs Program-controlled

Las computadoras digitales de tipo stored-program son aquellas que mantienen las instrucciones del programa y los datos en una memoria de lectura/escritura. Éstas fueron un avance sobre las program-controlled de los 40s, como la Colossus o la ENIAC, que eran programadas mediante enchufes e interruptores y no por un programa almacenado.

Las stored-program incluyen un set de instrucciones y puede guardar en memoria un conjunto de instrucciones que detallen los cálculos. La posibilidad de tratar las instrucciones como datos es lo que hace a los assemblers, compilers, linkers, loaders, y otras herramientas automáticas de programación posibles.

1.1.1. ISA (Instruction Set Architecture)

Es la parte de la arquitectura de computadores en relación con la programación, incluyendo los tipos nativos de datos, instrucciones, registros, modos de direccionamiento, arquitectura de memoria, manejo de excepciones e interrupciones y dispositivos externos de I/O.

Una ISA incluye una especificación del conjunto de códigos de operación (opcodes en el lenguaje de máquina, como assembly), y los comandos nativos implementados por un procesador en particular.

2. Input/Output

Es la comunicación entre la computadora y un humano u otro sistema de procesamiento de información. Inputs son las señales o datos recibidos por el sistema y outputs son las señales o datos enviados por éste. Por ejemplo, lectura de datos desde un disco externo es una operación de I/O.

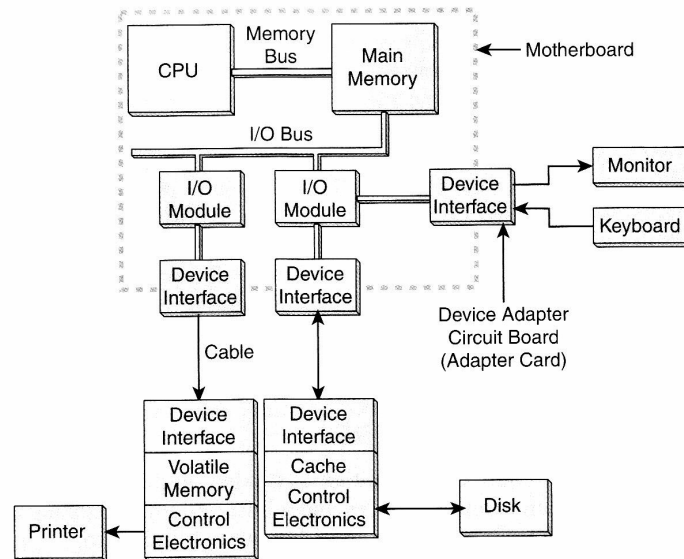


FIGURE 7.1 A Model I/O Configuration

El mapeo de I/O sirve para acceder al dispositivo, que puede ser:

En memoria a través de instrucciones de lectura y escritura en memoria .

En un espacio de I/O mediante instrucciones especiales.

Para comunicarse, los dispositivos de I/O cuenta con operaciones de polling o interrupciones:

Polled I/O aquel que es consultado periódicamente. Mientras no esté listo, la CPU vuelve retoma su actividad. Es sincrónica.

Interrupt-driven I/O aquel que inicia una IRQ (interrupt request) cada vez que requiere la atención del sistema. Internamente son implementados como señales electrónicas de alerta. Son asíncronas y pueden ocurrir en medio de la ejecución de una instrucción, lo que hace que se deba tener un cuidado extra al programar.

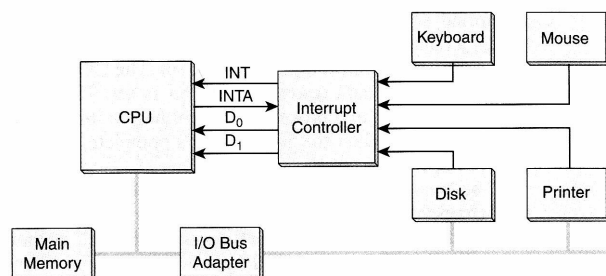


FIGURE 7.2 An I/O Subsystem Using Interrupts

Algunos microprocesadores cuentan con registros que funcionan como IMR (Interrupt Mask Register), como el Motorola 6800 que dentro de su SR (Status Register) los bits 10 al 8 se utilizan como máscara de interrupciones. El IMR especifica qué interrupciones serán ignoradas y no reconocidas (por ende, cuáles serán atendidas). Esta distinción permite diferenciar una que espera reconocimiento y otra que espera EOI (fin de la

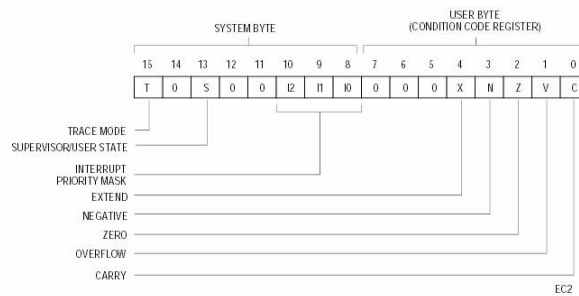


Figure 4-2. Status Register

interrupción). Esto sólo sirve para dispositivos enmascarables (los piden interrupción con la señal INTR). Los que son NMI (non-maskable interrupts) tienen mayor prioridad ante los enmascarables.

El **acceso directo a memoria (DMA)** es una propiedad que permite a ciertos subsistemas de hardware acceder a la memoria principal del sistema (RAM) independientemente de la CPU. Con DMA, la CPU primero inicia la transferencia, luego hace otras operaciones mientras la transferencia está en proceso, y finalmente recibe una interrupción desde el controlador de DMA cuando la operación finaliza. Esto es útil cuando la CPU no puede mantener el data transfer rate o cuando el CPU necesita performar en otras operaciones mientras espera una transferencia lenta de datos. También es utilizada para lo copia o la movilización de datos de memoria a memoria. Algunos dispositivos que utilicen DMA son: controladoras de discos rígidos, tarjetas gráficas, tarjetas de red y tarjetas de sonido.

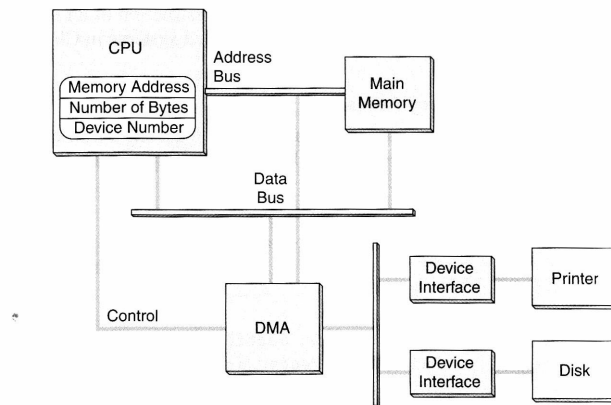


FIGURE 7.5 A Sample DMA Configuration

2.1. Hardware Interrupts

1. El controlador del dispositivo de I/O activa la señal de interrupción solicitada.
2. Cuando la CPU termina de ejecutar la instrucción en curso, verifica si hay interrupciones pendientes. Activa la línea de reconocimiento de interrupción (si la posee).
3. Detecta quién la interrumpió (autovectorización -soft- o vectores de interrupción -hard-)
4. Guarda el contexto del programa en curso en la pila (PSW y PC)
5. Deshabilita las interrupciones (Global -único nivel, interno, Intel- o Selectivo -multinivel, interno, Motorola)
6. Coloca en PC la dirección de la RAI a utilizar que obtuvo de la tabla de vectores de interrupción (avec o vect. int. ext.)

2.2. Software Interrupts

1. Si utilizó autovectorización, se detecta el dispositivo que originó la interrupción y se la reconoce seteando algún registro interno dle controlador de dispositivos de I/O.

2. Habilita las interrupciones (primero selectivo externo, guardando la máscara previa -si posee-, segundo global interno).
3. Inicia la rutina específica del dispositivo.
4. Deshabilita las interrupciones (primero global interno, segundo selectivo global restaurando la máscara previa).
5. Retorna de la interrupción con una instrucción que hace todo por hardware (RTI o IRET), restaurando el PC de la pila y el PSW de la pila (datos de interrupción).

Método I/O	Complejidad hardware	Complejidad Software	Velocidad
Polling	+	+++	+
Interrupciones	++	++	++
DMA	+++	+	+++

Cuadro 1: Conclusiones

3. Buses

Un bus es un camino de comunicación entre dos o más dispositivos. Es un medio de transmisión compartido y un medio de control.

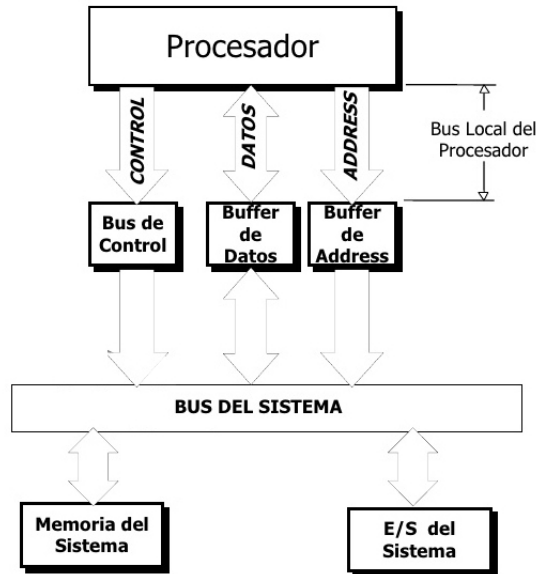


Figura 1: Estructura de bus clásica

3.1. Diseño de Bus

3.1.1. Tipo de líneas

Líneas dedicadas

- Dedicación física: conectan siempre le mismo subconjunto de módulos (Ej.: bus de dispositivos de I/O)
- Dedicación funcional: realizan siempre la misma tarea (Ej.: líneas de control en cualquier bus).

Ventaja hay menos disputas por el acceso al bus.

Desventaja incrementa en tamaño y precio.

Transferencia de datos en un bus dedicado:

- Escritura (master a slave): en un ciclo de clock master envía la dirección + datos por buses distintos.
- Lectura (slave a master): en un ciclo de clock master envía la dirección por bus de direcciones+slave coloca el dato en el bus de datos.

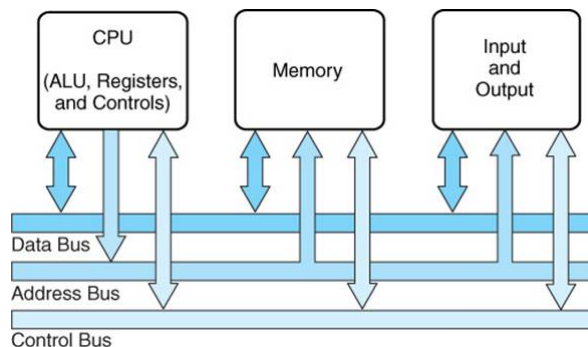


Figura 2: High Level I/O - System Bus (Líneas dedicadas)

Líneas multiplexadas

- Propósitos diferentes en distintos instantes de tiempo (Ej.: bus de datos / dirección según una línea de control).

Ventaja hay menos líneas, lo que reduce tamaño y precio.

Desventaja se complica la circuitería, reduciendo la velocidad del computador.

Transferencia de datos en un bus multiplexado:

- Escritura: transmisión de dirección + transmisión de dato
- Lectura transmisión de dirección + espera que slave coloque dato (transferencia de bloques de datos: dirección + varios ciclos de datos)

3.1.2. Ancho del Bus

El ancho se define por el número de líneas del bus. Afecta directamente el desempeño del sistema.

- Ancho del bus de datos \Rightarrow nro. de accesos a memoria
- Ancho del bus de direcciones \Rightarrow cantidad de direcciones

3.1.3. Temporización

Se trata de la coordinación de eventos en el bus.

Sincrónica Incluye reloj.

- **Ventaja:** facilidad de implementación y testing.
- **Desventaja:** velocidad de reloj se adecúa al más lento.

Asíncrona Los eventos que suceden provocan nuevos eventos.

- **Ventaja:** mejora el rendimiento cuando hay dispositivos lentos y rápidos.
- **Desventaja:** difícil de implementar.

3.2. Arbitraje

Los dispositivos conectados a un bus necesitan control para realizar algunas acciones, por ej., cuando la CPU necesita un dato de memoria o un device de I/O necesita leer/escribir dato en memoria sin pasar por la CPU.

El control del bus secuencial maneja un dispositivo a la vez.

Centralizado necesita controlador de bus o árbitro (se usa un chip o una parte de la CPU).

Distribuído cada módulo incluye un sistema de control de acceso y entre todos controlan el bus.

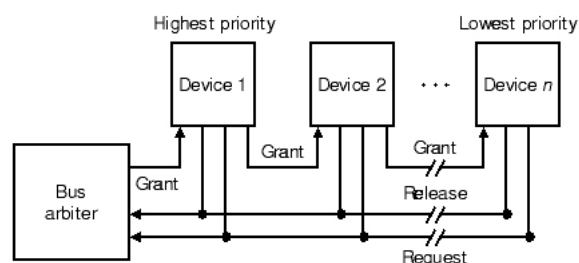


Figura 3: Bus arbiter

3.3. Conexiones

Serie se envían los datos un bit a la vez, de manera secuencial.

1. USB (Universal Serial Bus - Periféricos)
 - modelo de transferencia high speed: 480 MB/s (USB 2.0)
2. Ethernet, Fast Ethernet, Token Ring (conexiones de red)
3. Firewire (interconexión de ordenadores y periféricos para aplicaciones multimedia)
4. Bluetooth (conexión entre PCs, móviles y dispositivos portátiles)
5. 802.11 wireless LAN (redes LAN sin cables)

Paralelo se envían múltiples bits simultáneamente.

1. ISA - EISA ((Extended)Industrial Standard Architecture)
 - bus de IBM, se incluye generalmente por compatibilidad Intel
 - permite interconexión con otros buses
 - 8.33 MB/s
2. PCI (Peripheral Component Interconnect)
 - permite interconexión con otros buses
 - 528 MB/s
3. AGP (Accelerated Graphics Port)
 - bus dedicado de alta velocidad (clock del bus de la CPU) y alto rendimiento para controlador gráfico.
 - 528 MB/s ó 1 GB/s
4. IDE (Integrated Drive Electronics - HDD, CD-Rom, DVD)
 - integrada en placas base
 - costo reducido
 - PATA 133 MB/s
 - SATA 150 MB/s
5. IEEE 1284 (Impresoras y escáners)
 - SPP (Standard Parallel Port - primer standard bidireccional)
 - EPP (Enhanced Parallel Port) 2 MB/s
 - ECP (Extended Capabilities Port) 5 MB/s

4. Memorias

Son dispositivos utilizados para el almacenamiento de información.

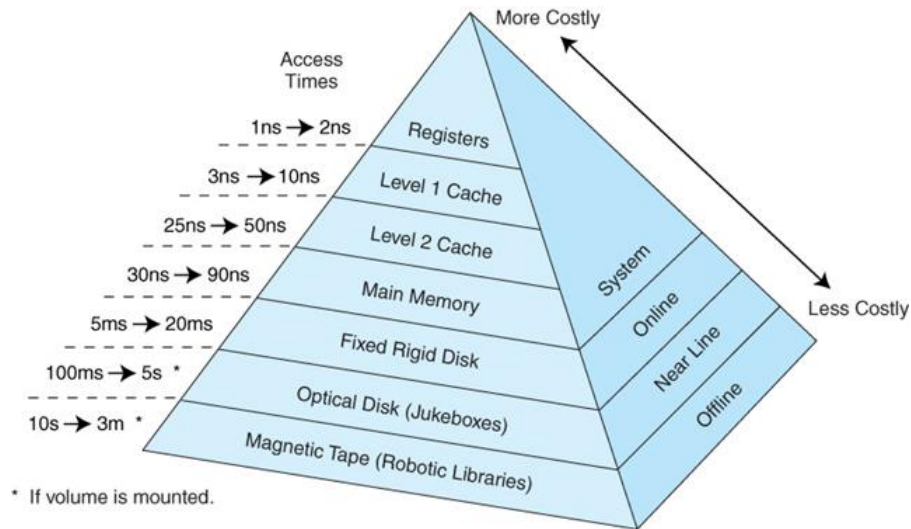


Figura 4: Jerarquía de las memorias

Métricas de las memorias:

- Capacidad de almacenamiento: bytes o múltiplos (kB,MB,TB)
- Tiempo de acceso: segundos o submúltiplos (ns, ms)
- Velocidad de transferencia de datos: en bytes/seg o múltiplos
- Consumo de energía : Watts
- Tamaño físico: cm³

4.1. Tipos y tecnología de memorias

Volátiles Aquellas que requieren energía para mantener la información almacenada; conserva sus contenidos mientras está encendida y cuando se apaga los datos se pierden inmediatamente o muy rápidamente.

1. Memorias SRAM (Static Random Access Memory)

- Almacenan la información en un biestable (flip-flop o latch).
 - Son costosas pues cada celda (1 bit) se compone de seis transistores ⇒ menor capacidad de almacenamiento por chip ⇒ **alto costo por bit**.
 - 3 transistores consumen energía máxima en forma permanente y los otros 3 consumen mínima energía ⇒ **alto consumo relativo**.
 - La lectura es directa y no destructiva ⇒ **rápido acceso**.
 - Se usan para formar la memoria caché.
- ⇒ si construimos el banco de memoria utilizando SRAM el costo y consumo de la computadora son altos.

2. Memorias DRAM (Dynamic RAM)

- Almacenan la información como una carga en un transistor.
 - Una celda (1 bit) se implementa con un único transistor ⇒ máxima capacidad de almacenamiento por chip ⇒ **bajo costo por bit**.
 - Cada transistor consume mínima energía ⇒ **consumo mínimo**.
 - Al leer el bit se descarga la capacidad, por lo que necesita regenerar la carga ⇒ **considerable tiempo de acceso**.
- ⇒ si construimos el banco de memoria utilizando DRAM no se aprovecha la velocidad del procesador.

No Volátiles Aquellas que pueden recuperar la información almacenada, incluso después de haber sido desconectadas.

1. Memorias ROM (Read-Only Memory)

Los datos almacenados sólo pueden modificarse lentamente, con dificultad, o no del todo, por lo que se utiliza principalmente para distribuir el firmware (software que está muy estrechamente ligado a hardware específico, y es improbable que necesite actualizaciones frecuentes).

2. Memorias PROM, EPROM y EEPROM

- PROM (Programmable Read-Only Memory)

Se utilizan en dispositivos electrónicos digitales para almacenar datos o programas permanentes, por lo general bajo nivel como el firmware. La diferencia con una ROM estándar es que en una PROM los datos se programan después de la fabricación.

- EPROM (Erasable Programmable Read-Only Memory)

Se pueden programar una única vez.

- EEPROM (Electrically Erasable Programmable Read-Only Memory)

La diferencia entre EPROM y EEPROM radica en la forma en que los programas de memoria y borra. EEPROM se puede programar y borrar eléctricamente.

3. Memorias Flash

La memoria flash se desarrolló a partir EEPROM. Hay dos tipos principales, que llevan el nombre de las compuertas NAND y NOR lógicas puesto que las celdas de memoria flash individuales exhiben características internas similares a las de las compuertas correspondientes.

5. Memorias caché

Son bancos de SRAM más pequeños y de muy alta velocidad que contienen una copia de los datos e instrucciones que están en la memoria principal. La CPU los utiliza para reducir el tiempo de acceso estos items en principal sin recurrir a wait states (esperar que la memoria esté lista para el acceso (ready)).

Requiere de hardware adicional que asegure que la caché contenga los datos e instrucciones más frecuentemente utilizados por el procesador.

La mayor parte de las CPUs tienen cachés distintas e independientes, incluyendo caché de instrucciones y caché de datos donde, en general, la caché de datos está organizada en niveles jerárquicos (L1, L2, etc.).

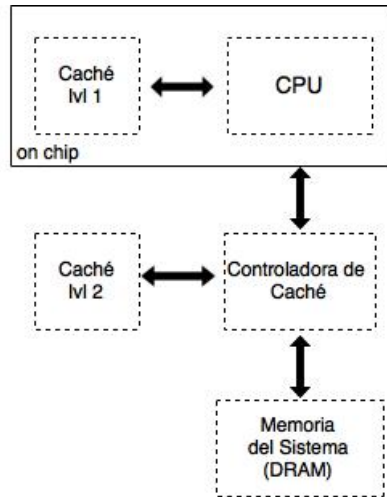


Figura 5: Caché Multinivel : size lvl 2 cache > size lvl 1 cache , speed lvl 1 cache > speed lvl 2 cache

Tamaño: Debe ser lo suficientemente grande para que el procesador resuelva la mayor cantidad posible de búsquedas de código y datos en esta memoria asegurando alta performance y lo suficientemente pequeña para no afectar el consumo y costo del sistema.

5.1. Performance

Se logra un hit cuando se accede a un ítem (dato o código) y éste se encuentra en la memoria caché. En caso contrario se considera miss.

La proporción de accesos que resultan en hit pueden ser una buena medida de la efectividad de la caché dado un programa o algoritmo. Se espera un hit rate lo más alto posible para una buena performance (hit rate = $\frac{\#hits}{\#accesos\ totales}$)

5.1.1. Impacto de un caché miss

El **pipeline** permite superponer en el tiempo la ejecución de varias instrucciones a la vez. No requiere hardware adicional, sino lograr que todas las partes del procesador trabajen a la vez. Trabaja con el concepto de una línea de montaje: cada operación se descompone en partes y se ejecutan en un mismo momento diferentes partes (stages) de diferentes operaciones.

Si la búsqueda de una instrucción o un operando falla, entonces el procesador debe recurrir a la memoria principal. La demora en el acceso hace que el pipeline se atasque (stall). Una vez recuperado el dato de memoria, se requieren varios ciclos de clock para recuperar el ritmo de operación del pipeline.

5.2. Controlador de caché

El controlador de caché trabaja mediante dos principios que surgen de analizar el comportamiento de algoritmos de software que se emplean habitualmente:

Principio de vecindad temporal Si un ítem es referenciado, la probabilidad de ser referenciado en el futuro inmediato es alta.

Principio de vecindad espacial Si un ítem es referenciado, es altamente probable que se referencia a los ítems vecinos a éste.

5.3. Coherencia

Una variable que está en caché también está alojada en alguna dirección de DRAM (ambos valores deben ser iguales). Cuando el procesador los modifica hay varios modos de actuar:

Write through El procesador escribe en la DRAM y el controlador de caché actualiza el caché con el nuevo dato.

Write through bufferd El procesador actualiza la SRAM caché y el controlador aché, luego actualiza la copia en memoria RAM mientras el procesador continúa ejecutando instrucciones y usando datos de la memoria caché.

Copy back Se marcan las líneas de la memoria caché cuando el procesador escribe en ellas. Luego en el momento de eliminar esa línea del caché el controlador caché deberá actualizar la copia de DRAM.

Si el procesador realiza un miss mientras el controlador de caché está accediendo a la DRAM para actualizar el valor, deberá esperar hasta que el controlador de caché termine la actualización para recibir desde éste la habilitación de las líneas de control.

5.4. Estructura de memoria caché

Línea Elemento mínimo de palabra de datos dentro de la caché. Corresponde a un múltiplo del tamaño de la palabra de datos de memoria. (Cuando se direcciona un item en memoria, generalmente se requerirá de los items que lo rodean - ppio de vecinidad espacial)

5.4.1. Organización

Mapecto directo utiliza Tag e Index.

Totalmente asociativa utiliza Tag, Line e Index.

Asociativa por conjuntos de n vías utiliza Tag, Set e Index.

5.4.2. Políticas de reemplazo

Para dar lugar para nuevas entradas dado un miss, la caché puede que tenga que desalojar una entrada existente. El problema fundamental con cualquier política de reemplazo es que tiene que poder predecir qué entrada en la caché es menos probable a usarse en el futuro próximo.

Algunos algoritmos populares de reemplazo del contenido de la memoria caché son:

- LRU (Least Recently Used) -ppio de vecinidad temporal-
- LFU (Least Frequently Used)
- FIFO (First In, First Out)
- Random

5.5. Implementaciones prácticas de memoria caché

- Intel 80486

Tamaño caché 8 KB lvl1

Tamaño line 16 B

Organización Asociativa de 4 vías

- Pentium (dos cachés, una para datos y otra para instrucciones)

Tamaño caché 8 KB c/u

Tamaño line 32 B

Organización Asociativa de 4 vías

- Power PC 601

Tamaño caché 32 KB

Tamaño line 32 B

Organización Asociativa de 8 vías

A partir de este ejemplo todas los modelos ejemplo a continuación cuentan con dos cachés: una para datos y otra para instrucciones.

- PowerPC 603

Tamaño caché 8 KB c/u

Tamaño line 32 B

Organización Asociativa de 2 vías

- PowerPC 604

Tamaño caché 16 KB c/u

Tamaño line 32 B

Organización Asociativa de 4 vías

- PowerPC 620

Tamaño caché 32 kB c/u

Tamaño line 64 B

Organización Asociativa de 8 vías